NETWORK MONITORING TRAFFIC COMPRESSION USING SINGULAR VALUE DECOMPOSITION

THESIS

Steven N. Feigh, Major, USA

AFIT-ENG-14-M-27

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-14-M-27

NETWORK MONITORING TRAFFIC COMPRESSION USING SINGULAR VALUE DECOMPOSITION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Cyber Operations

Steven N. Feigh, BS

Major, USA

March 2014

AFIT-ENG-14-M-27

NETWORK MONITORING TRAFFIC COMPRESSION USING SINGULAR VALUE DECOMPOSITION

Steven N. Feigh, BS

Major, USA

Approved:

_____//signed//_____	___11 March 2014___
Maj. Kennard R. Laviers, USAF, PhD (Chairman)	Date
_____//signed//_____	___11 March 2014___
LTC Robert J. McTasney, USA, PhD (Member)	Date
_____//signed//_____	__11 March 2014___
Kenneth M. Hopkinson, PhD (Member)	Date

AFIT-ENG-14-M-27

Abstract

With increasing magnitude of computer network activity, the ability to monitor all network traffic is becoming strained. The need to represent large amounts of data in smaller forms is essential to continued growth of network monitoring tools and network administrators' capabilities. Network monitoring captures many different measurements of the data flowing through the network. This thesis introduces a new method of sending network traffic monitoring data that reduces the overall volume of data from the traditional method of packet capture. By populating a matrix with specific data values in a sparse format, this experiment reduces the data using singular value decomposition (SVD) compression. Matrices were populated using network monitoring datasets from 1996 Information Exploration Shootout (IES). The data populated into the matrices was varied along time frame and data field to determine if the SVD compression algorithm reduced the quantity of original data values. Results indicated that the quantity of data varies dependent on the volume of the data field chosen. The matrix population method was based on port values to allow combining values within the matrix cells. The results trended to a successful reduction of data if the time frame is increased significantly. However, increases in the time frame led to less distinction of the individual data values.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Major Kennard Laviers, for his guidance, patience, and support throughout the course of this thesis effort. The insight and experience was greatly appreciated.

Steven N. Feigh

Table of Contents

Page

List of Figures

Page

List of Tables

Page

NETWORK MONITORING TRAFFIC COMPRESSION USING SINGULAR VALUE DECOMPOSITION

I. Introduction

1.1. Background

The world continues to become more and more connected with a growing number of portable devices connected to the internet. It is not uncommon for an individual to own multiple computers or computing devices. With this growing number of devises, computer networks are quickly growing large and more sophisticated. It is estimated that the volume of computing devises on a typical network has increased 12 times over the past 12 years [16]. In the last two decades, organizations such as the United States military have gone from a handful of computers per unit of hundreds of personnel to practically one computer per person.

The ability to monitor the security of these complex networks has become increasingly more difficult and often administrators do not notice attacks until it is too late [33]. Network administrators currently use a number of programs or tools to oversee their networks. Most of these tools either generate or read existing network logs and streams of data [26]. These logs are then interpreted in a number of ways for the administrators. Some tools display the raw data which can be searched or parsed based on queries for specific types of data or network packets [9, 23]. However, given the trends of increased traffic, the data these tools gather and search through are becoming too large to detect anomalies. Given these facts, it is apparent that a new method of network monitoring is needed.

Current monitoring systems are not sophisticated enough to filter only malicious activities and often generate too many alerts of potential threats for an administrator to handle manually [7]. Network traffic compression can reduce cost in administrator time and resources needed by compressing large data volumes into smaller ones. This research attempts to compress network data in such a way that when the data is needed, it still represents the same unique information as it did before it was compressed.

The use of singular value decomposition (SVD) compression is originally comes from research in digital image compression [3, 31]. Digital images are effectively data values in matrix form [25]. If these images can be compressed without loss of image quality and still reduce the amount of data, could a sparsely filled matrix of network traffic data do the same for raw network data?

1.2. Problem Statement

The size and scope of computer networks are continuously increasing. The volume of data a network administrator manages increases by a factor of three every year, according to Gilder's Law. The amount of network traffic each device sends and receives can vary greatly depending on device type, purpose, and frequency of use. Network monitoring tools use various parts and elements of network traffic to monitor and identify potential malicious activity. Each of these elements corresponds to a specific field or measurement of a network packet or network connection. Fields can vary depending on the monitoring software. Rarely is every data element needed to detect a specific type of attack or to perform a specific type of analysis [6]. A network monitoring tool only needs to analyze only specific elements of traffic data in order to

detect a certain type of attack, but the same tool will need to analyze different elements of traffic data in order to detect a different type of attack. If this specific data could be reduced, then the storage and the transmission of network monitoring data would also be reduced. The experiment conducted attempted to answer the following questions: How much can network traffic data be reduced by using SVD before the data values change? Is the size of the reduced decomposed data less than the size of the original using SVD compression?

1.3. Goals

The first goal is to determine how much a matrix containing a particular element of network traffic can be reduced and still retain the same values of the original data. Using single elements of network monitoring traffic populated into a matrix should allow a reduction of at least 50% from original size of the matrix decompositions and still retain the same number of values in the same fields of the matrix. This hypothesis is based on the principles of singular value decomposition [31]. The one of the decomposition matrices contains the singular values of the original matrix and allows the values to be ordered. By ordering these values from greatest to least, it allows for the least important values to be dropped. Therefore eliminating least significant data reduces the original matrix.

The second goal is to determine if the resulting maximum reduced decompositions from the original matrix contain less overall data values than the original dataset. This is an indirect comparison. The original quantity of data values is counted by multiplying the volume of packets by the number of fields used for each packet over

the measured time frame. The compressed data consists of the number of non-zero values contained in the reduced decomposition matrices from SVD compression. The non-zero values are used because of how sparse matrices can be stored and transmitted.

1.4. Hypothesis

Singular value decomposition compression was originally used as a form of image compression [3, 31]. Since images are just complex matrices that are interpreted visually, if an image's data size can be reduced significantly, without much visual distortion, the same should be true for a standard matrix. This leads to the hypothesis that a single matrix populated with a single field of data from network monitoring traffic can be reduced using SVD compression such that the amount of data is less than the amount used to create the original matrix.

1.5. Scope

This research addresses the need to develop a method of representing the vast pools of data gathered from networks in a structure that can be easily compressed to save system resources for either transport or storage. The research in this thesis is centered on the larger problem how to handle the amount of data generated and passed through a network. The specific data that is used to monitor a computer network varies between network monitoring tools. The scope of this thesis is to analyze the data packets measured from a simulated computer network dataset. This thesis focused on the individual fields and values of these network packets, in order to not restrict the research relevance to a specific network monitoring tool.. Maintaining the values during experimentation was priority to allow any monitoring tool to have access to the original

data. This aspect aided in determining compression and overall reduction of values as a result of the SVD compression process.

1.6. Assumptions and Limitations

The virtual environment where all calculations are preformed is MATLAB version 2012b. An assumption is the calculations invoked for the singular value decomposition process are preformed correctly in the MATLAB environment. During the compression algorithm, the program invokes the 'svd( )' command to decompose the original matrix into its separate decomposition parts. The 'eig( )' function that produces the eigenvalues of a given matrix is also used in this experiment to evaluate the original matrices. Given MATLAB's reputation as an environment used for a multitude of research experiments in various academic fields, these assumptions are minimized for risk of miscalculation [5, 24].

This research is limited to the network data collected in the Information Exploration Shootout. This dataset is explained in detail in Chapter III. The data set contains a record of network packets used to simulate computer network traffic. The data used contains five datasets, each with thirteen specific element fields of data corresponding to an individual packet. These datasets were collected in 1996, and the experiment system is limited to the data fields and thresholds of a network from that time. Modern and larger networks could create a larger variety of fields to be analyzed. However, resources available and time constraints of this experiment prevented the generation of newer network traffic data.

1.7. Methodology

Current network monitoring tools collect specific data fields from the network to include: duration of connections, protocol type, destination service, packet size, direction of travel and several others [5, 10, 26]. The method introduced in this body of work selects the specific data fields from the provided datasets, and parses them into a fixed NxN matrix. The cells of individual matrices contain the values of the data field chosen over the measured time frame. After the matrix is populated for a time frame, a new matrix is created from the original using SVD compression. The quantity of values from the resulting decomposition matrices is compared to the quantity of original data used to create the matrix. Comparisons were made between the original matrices and the compressed matrices by a cell-by-cell comparison of values. These comparisons were in the calculated matrix to ensure data integrity is not lost during the compression process.

1.8. Implications

Data storage and transportation resources could potentially be reduced by using SVD compression of individual network data elements over time. If shown to be beneficial, a network traffic compression algorithm would allow for a central repository of original data to be maintained and only the required data be transferred for analysis. The transfer would require less bandwidth from point to point and simple matrix math to reconstruct the original data at the destination. A reduction in volume of data would facilitate the use of mobile devices in the support of network monitoring where system resources are limited in comparison to standard computer systems.

II. Literature Review

The purpose of network monitoring is to maintain situational awareness of activities on a digital network. Network administrators use numerous tools to determine what is connected to their network, and other tools to monitor how their network is behaving. The current state of network monitoring relies on event logs and automated systems to maintain the network [20]. This reliance does not diminish the prevalent challenges of the volume of alerts being too numerous for manual analysis. Any change to a network can create undesired affects resulting in false positives showing up on current monitoring tools.

2.1. Network Monitoring Tools

Most network traffic is normal and nonthreatening to the network. However, discovering a small string of malicious packets among the potentially millions sent and received daily is equivalent to finding a needle in a haystack. Manual sorting is arduous, time consuming, and prone to human error. Intrusion detection systems (IDS) generate logs that capture information needed to identify malicious activity [15, 21, 26, 28]. Intrusion detection systems allow a low number of trained administrators to monitor vast networks. Despite the availability of IDS, the problem remains of sorting through these extensive security logs [15]. While the needs for an IDS can vary greatly depending on the purpose and scope of a network, the needs for an IDS focused on are real-time monitoring and attack pattern analysis.

2.1.1. Real – Time traffic monitoring

The ability to track network traffic is the basis of network monitoring. There are two ways of representing traffic across the network. The first method is the use of individual packet data. Capture tools like Wireshark [23] and NAM [9] show every bit of data from each individual packet as it is transferred across the network. Large volumes of information are great for analysis of the data being sent and received. However, this data requires proportionally large data storage to accommodate the volume of traffic on the network.

The second method of network monitoring is the use of flows. Flows are created by using a construct which combines the individual packets from the same sessions that summarize the communication between two IP addresses or ports over time [4]. Moving extensive amounts of data becomes less of a burden as network bandwidth has grown larger. As a result of the growing volume of data, packet animation is used less often than flows for IDS [4, 9, 13].

Network security is inherently time-sensitive. The quicker an administrator or security analyst can react to a threat, the more likely a threat can be mitigated. The goal of most analysis is to analyze data as soon after collection as possible. Since not every network flow needs to be reviewed, a broad awareness allows analysts to perceive current activities. In a real-time environment, noticing changes allows for quicker detection of a potential threat by an administrator. System resources or personnel are then deployed to perform appropriate remedies to address the issue.

2.1.2. Attack Pattern Analysis

Attack pattern analysis is instrumental to detect parallel coordinated attack vectors (PCAV) [6, 11, 19, 26]. A monitoring tool has many ways to represent PCAV so new attacks are quickly detected and enable network administrators to recognize and respond to those attacks. Analyses of attack characteristics help determine the visual mechanism for representing the most popular internet attacks: DDoS attacks, worm attacks, or network scans. These attacks have one common characteristic, the one-to-many ratio between the attackers and the victims, whereas legitimate flows tend to have a one-to-one relationship. Four parameters that have been identified as useful indicators of attacks occurring on networks are shown below.

1. The source IP address and destination IP address in flow information are selected as parameters because they specify the attacker and the victim host.
2. Destination port number is selected as a parameter. This value identifies the targeted service of an attack and verifies port scanning attacks.
3. The average size of packets in a flow can be used as a parameter that gives some clues whether the flow is suspicious or not. Even when the packets have payloads, usually the length of packets is fixed.
4. The use of transmission control protocol flags in the header portion of network packets and the protocol field in IP headers are another parameter. These flags indicate the need for specific action of attention of the system.

These parameters were instrumental in discovering a suitable dataset to perform this experiment.

2.2. Current Issues with Network Monitoring Tools

Current monitoring systems are not sophisticated enough to detect all malicious activities and often generate many alerts for an administrator to process. A vast quantity of alerts requires network administrators to manually sort through them [26]. The three most prominent issues facing network monitoring tools are the need for multiple tools, form and function, and scalability.

2.2.1. Need for Multiple Tools

A limitation of the tools currently in use is that they rarely employ multiple data sources and often require a separate application to parse the security data before it can be graphed or otherwise manipulated. Multiple applications can be inconvenient for the user and makes evaluating security events in a timely fashion difficult. Additional tools increase resource demands of the system by having multiple tools running simultaneously [29]. While these tools do accomplish their designed task, having multiple tools operating at the same time increases the requirement of base knowledge and skills on an administrator.

2.2.2. Form and Function

Security tools are written either by people who are network security personnel and have limited knowledge of visualization theory and human computer interaction, or by individuals with expertise in the field visualization and data representation who are not experts in the field of computer security. As a result, most current tools suffer from a lack of domain knowledge in one of these areas [35]. An important goal for any future

security tool is to close this gap and create an application that is user-friendly, technically accurate, and effective from a security perspective.

2.2.3. Scalability

As technology increases and the ability to process larger amounts of data increases along with it, scalability continues to become an area of concern when monitoring networks. However, sometimes the growth in data far exceeds the growth in computing power. Also, different ways of aggregating data are used to interpret massive amounts of data, but are not without drawbacks [1, 13]. Effective use of aggregation largely depends on the skills of the administrator requesting the information. If the application merely provides the opportunity for users to aggregate fields as they desire, less experienced administrators perform data requests that overload the target system [30]. Aggregation assumes that the end-user knows what he or she is looking for; and the end-user knows how and when to apply criteria to reduce the size of the result set from a given data request.

Visualization of network monitoring data is the most popular form of interpreting large amounts of data. Additionally, different techniques have been proposed for scaling up data interpretation in the form of visualizations [2,4,6,7,9,10,12,21,22,28,32].

2.3. Visualizing the Data

A technique for increasing an administrator's network monitoring capability is the visualization of captured network log data [21]. Visually analyzing data helps network administrators perceive patterns, trends, structures, and exceptions in complex data sources [28] thus decreasing the time to manually review multiple log files. Visualizing

log files allows network administrators to better recognize abnormal behavior by recognizing pattern differences at a glance. Visualizations do not provide a total solution and can be altered by data received from other network management software. Information overload, counter intuitive interfaces, and lack of actionable information can influence the visualization of the data causing the administrator to make less than optimal decisions when interpreting the visualized data [6].

A brief overview of the general practice of visualization design and the development of visual tools is included in several works [22,28,32]. Each of these overviews describes a variation to the four basic steps.

1. Input data is read in and stored in a standardized form.
2. The stored data is transformed to symbols. These symbols vary based on components of the data or how the data is interpreted.
3. The symbols are displayed on the screen using the computing device's rendering and display system.
4. The administrator visually processes the visualization of the data with an understanding of the data in the data set.

The administrator is then able to interact with the network according to what he or she perceives. Steps 2-4 are subject to distortion of the data and lack of a direct connection between the data and the perceived understanding of the data. The completed design of the visualization must take all of these components into account.

2.4. Visual Compression

Visualization is becoming the most popular method of representing the great quantities of data [3,15,20,21,32]. Visualization is not the focus of this thesis. However, research in this area lead to the concept of compressed visual images representing network traffic. Images are nothing more than matrices that are interpreted in a visual manner [25]. Singular value decomposition compression is an image reduction method that reduces the matrix data that comprise a digital image. The method of SVD compression does not require a conversion of the data into an image format but only a matrix format for implementation.

2.4.1. SVD Compression

Singular value decomposition can be looked at from three mutually compatible points of view. First, it can be viewed as a method for transforming correlated variables into a set of uncorrelated ones that better expose the various relationships among the original data items. At the same time, SVD is a method for identifying and ordering the dimensions along which data points exhibit the most variation. This leads to the third way of viewing SVD, as a way to find the best approximation of the original data points using fewer dimensions [3]. This leads to the assertion that SVD can be used as a method for data reduction. SVD is chosen as the form of compression for its ease in calculation and the evaluation of singular values, when calculating the decomposition. The singular values are the square root of the eigenvalues.

By applying SVD to the matrix, the matrix can be expressed as:

$$f = U\Sigma V^T \tag{1}$$

Where:

f = matrix of network element data
U = is an m_{rows} by m_{rows} orthogonal unitary matrix
Σ = is an m_{rows} by $m_{columns}$ diagonal matrix (0 except on its main diagonal)
V = is a $m_{columns}$ by $m_{columns}$ orthogonal unitary matrix, where $V^T = V^{-1}$ or transpose matrix of V

In the decomposition the diagonal entries of Σ are the singular values of f, and by convention they can be ordered by decreasing magnitude.

Since the network traffic elements are now in an NxN matrix format. The matrix can be treated much like an image matrix for compression. The matrix can be compressed by SVD by defining approximations f_s to f by

$$f_s = \sum_{i=1}^{s} u_i \sigma_i v_i^T \tag{2}$$

Where:

$\sigma = s$ largest singular values, $\sigma_1, \ldots, \sigma_s$, replacing the rest with zeros.
$u = s$ columns, $u_1, \ldots, u_s$, replacing the rest with zeros.
$v = s$ rows, $v_1, \ldots, v_s$, replacing the rest with zeros.

The largest singular values correspond to the most important information of the matrix. By the Eckart-Young Theorem [8], f_s is the best rank s approximation to f in the sense of minimizing:

$$\sum_{i,j=1}^{s} (f_{i,j} - g_{i,j})^2 \tag{3}$$

Where:

g = all matrices having exactly s nonzero singular values. Noting that with only s nonzero singular values of f_s, it is only necessary to store the first s columns of

U and s rows of V^T in order to represent f_s. The total number of elements needed to store is $s(1+m_r+m_c)$, which is less than $m_r m_c$ for $s < \dfrac{m_r m_c}{m_r m_c + 1}$.

Figure 1 illustrates the method of SVD compression with the example of a simple matrix, shown below.

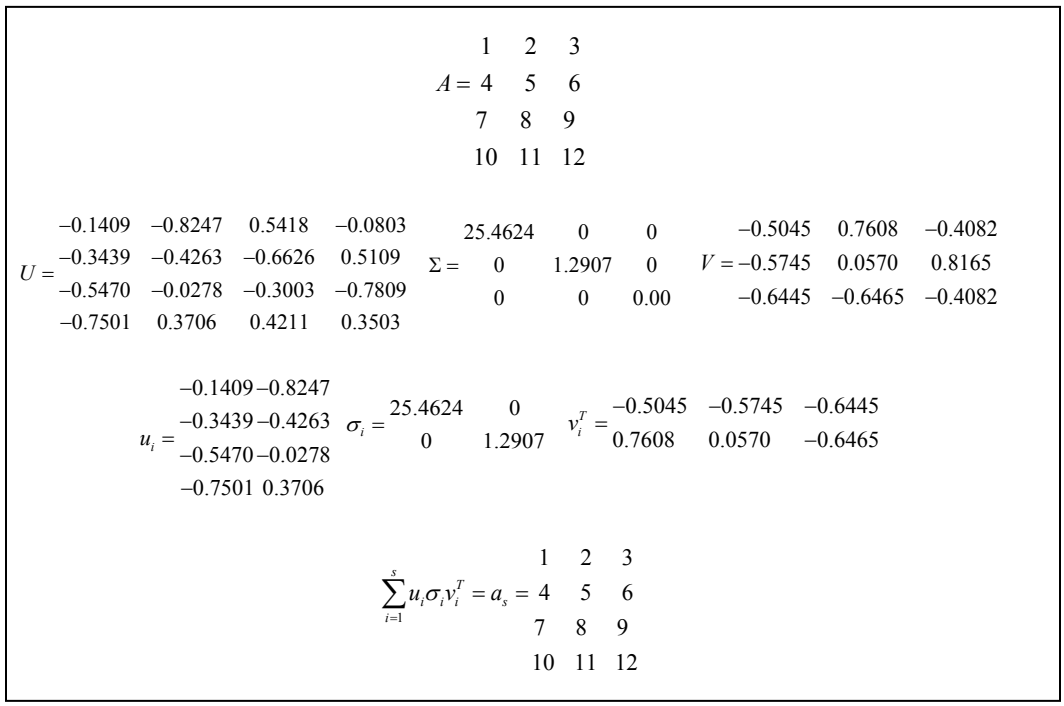

Figure 1. SVD Compression Example

III. Methodology

From the background research, monitoring tools are using various methods to represent large amounts of data to save system resources. One method taken from image compression is SVD compression. Testing SVD compression in relation to network monitoring traffic is the focus of the following experiment.

3.1. Approach

The proposed approach takes selected data elements from network monitoring traffic, and applies them to an NxN matrix. The cells in the matrix represent the specified network traffic monitoring field values gathered from datasets during the selected time frame of examination. Four time frames are tested in order to evaluate if data saturation affects the compression process. Each matrix is kept at a constant size, 200x200, in order to maintain consistency for placement of network traffic field value.

For this experiment, the source and destination port value is used for placement in the matrix. Port values from the source field are used for placement in individual rows, while destination port values are used to determine column placement in the matrix. The matrix is organized starting from the top left of the matrix. Values from packets with low numbered ports are located closer towards this area of the matrix as shown in Figure 2. As port values increase the position of the value to be placed move, to the right for increased destination port and farther down for increased source port.

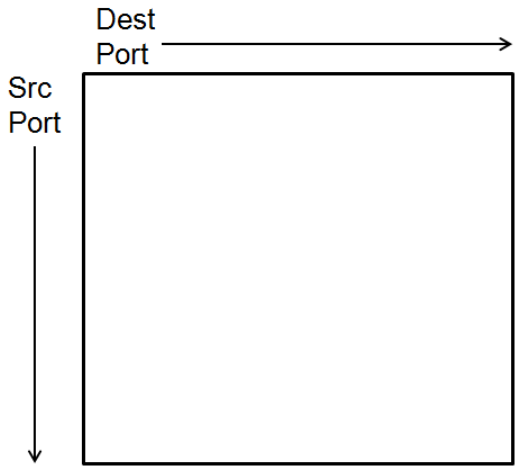

Figure 2. Matrix Cell Positioning

The nature of populating the matrices allows for the possibility of a matrix cell containing multiple values over a single time frame. The fields chosen for this experiment represent byte values captured from packets in the datasets. These values were chosen so they can be added over the time frame. The time frames used for each matrix were small to minimize the amount of data value combining.

The original matrices are reduced using singular value decompression compression. In order to assist in determining the maximum amount of SVD compression, the eigenvalues of each matrix are analyzed by quantity and value. Keeping only the most significant eigenvalues until Maximum compression is achieved. The Maximum compression is the amount of reduction to each of the decomposition matrices without causing a change in the values of the matrix by more than one. The number of values in the three matrix decompositions needed to calculate the new compressed matrix is compared to the number of values needed to create the original matrix. If the number of values from SVD compression is less than the number of values from original data, then the compression is considered successful.

3.2. System Boundaries

As shown in Figure 3 below, the System Under Test (SUT) is completely contained in a virtual environment, for the purposes of this experiment all algorithms, programs and calculations were performed in MATLAB 2012b on a Microsoft Windows 7 operating system environment.

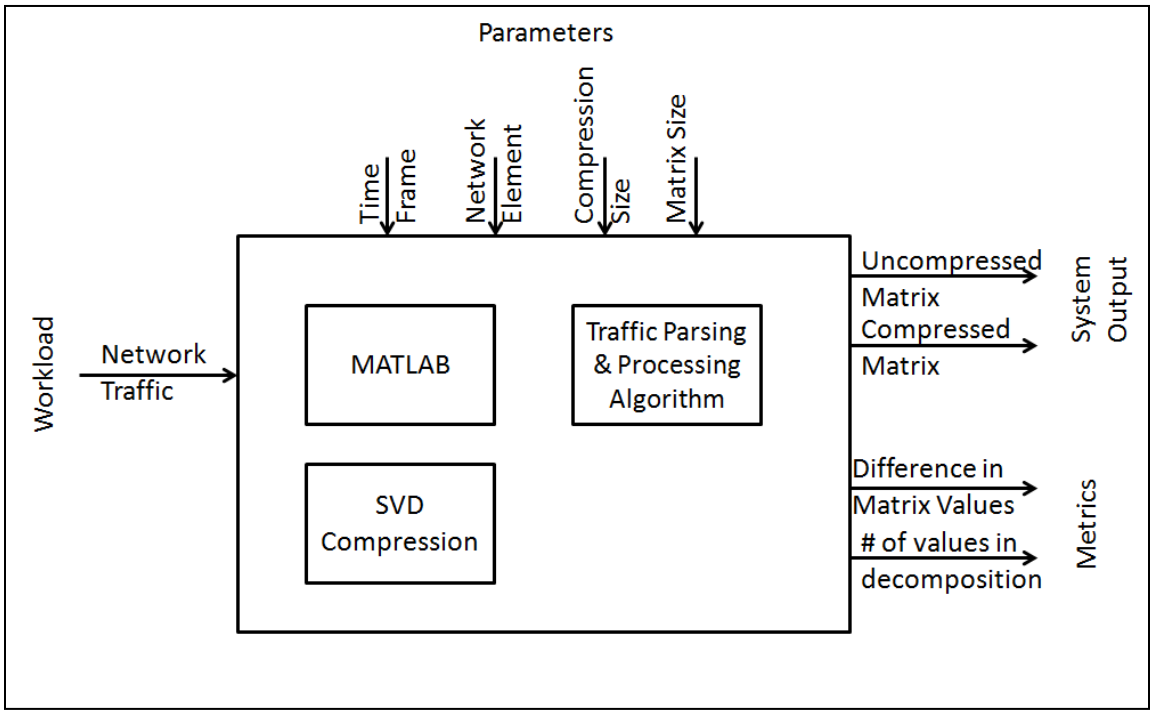

Figure 3. System Block Diagram

For matrix creation, the Information Exploration Shootout (IES) dataset was used. There are five different datasets of network traffic in IES. Each contains approximately twenty minutes worth of network traffic capture and varies with the number of lines of data, from approximately 350,000 lines in the baseline dataset to over 600,000 lines of network packet data in the largest dataset capture. These datasets were chosen because they provide a baseline, or 'clean' dataset, with no network attacks, as well as four other datasets that only contain a singular network attack. While the network attack itself is not

being examined during this experiment, the network attack is useful to determine the compression algorithms ability to handle the additional data. A key requirement is to not alter the original information in order to allow a network monitoring tool or administrator use it for analysis.

However, the IES dataset was created in 1996 for public use in discovering common network attacks by analyzing very basic network TCP/UDP log capture data [34]. Because the dataset is so old, the experiment is limited to the fields provided in the datasets. Newer, publicly available datasets that have been vetted by other research analysts are difficult to find. Most notable is the Knowledge Discovery and Data Mining, KDD'99, dataset created by Lincoln Laboratories for the Defense Advanced Research Projects Agency, DARPA [36]. KDD'99 dataset is widely used for network intrusion and network monitoring analysis, but did not meet the criteria for this experiment because the KDD'99 is missing time stamp field need to gauge time frame. Time frame is important for this experiment. The volume of network traffic packets changes over time dependent on network activity. The time stamp is used to assists this experiment in the simulation of real-time network data capture environment.

The Component Under Test (CUT) is the SVD compression algorithm given the presented matrix format. For the purposes of this experiment only three data fields are used in the compression algorithm. These fields were chosen because they represent byte value and can be combined if the values were to occupy the same position in the matrix. Combining values can potentially limit analysis of the data by obscuring the difference in packet data over too long of a time frame. However, sampling the network datasets at small intervals minimizes data combining.

3.3. Workload

The network traffic dataset provided for this experiment is the dataset created for the Information Exploration Shootout (IES) in 1996 [11]. IES dataset consists of five individual network traffic files containing a baseline set of network traffic that contains no attack traffic data and four with a single type of network attack each. These attack types are listed in Table 1.

Table 1. Network Datasets Attacks

Name	Type of Attack
Baseline	None
Network 1	IP spoofing
Network 2	FTP password guessing
Network 3	Network Scanning
Network 4	Network Hopping

The traffic was captured using the tcpdump tool and contains only thirteen different fields or elements for each packet that passed through the network. Each file is in comma separated value format and contains approximately 20 minutes of traffic captured on the network [32]. All source and destination Internet Protocol (IP) addresses where masked to keep the network topology from being revealed. As seen in Appendix A, the IP addresses where limited to only the fourth octet of digits. Additionally, an initial analysis of each dataset shows that each packet either contains a source or destination an address with the final octet with a value of "2".

Through the initial analysis of the baseline profile, the following information about the traffic of the network was confirmed [14]. Most of the baseline request is composed of http requests, port 80, more frequently outbound than inbound. The smtp protocol, port 25, is also common on the network. Also many connections occurred from

a few external hosts to UDP port 7001. This illustrated a histogram plot of the Baseline dataset in Figure 4.

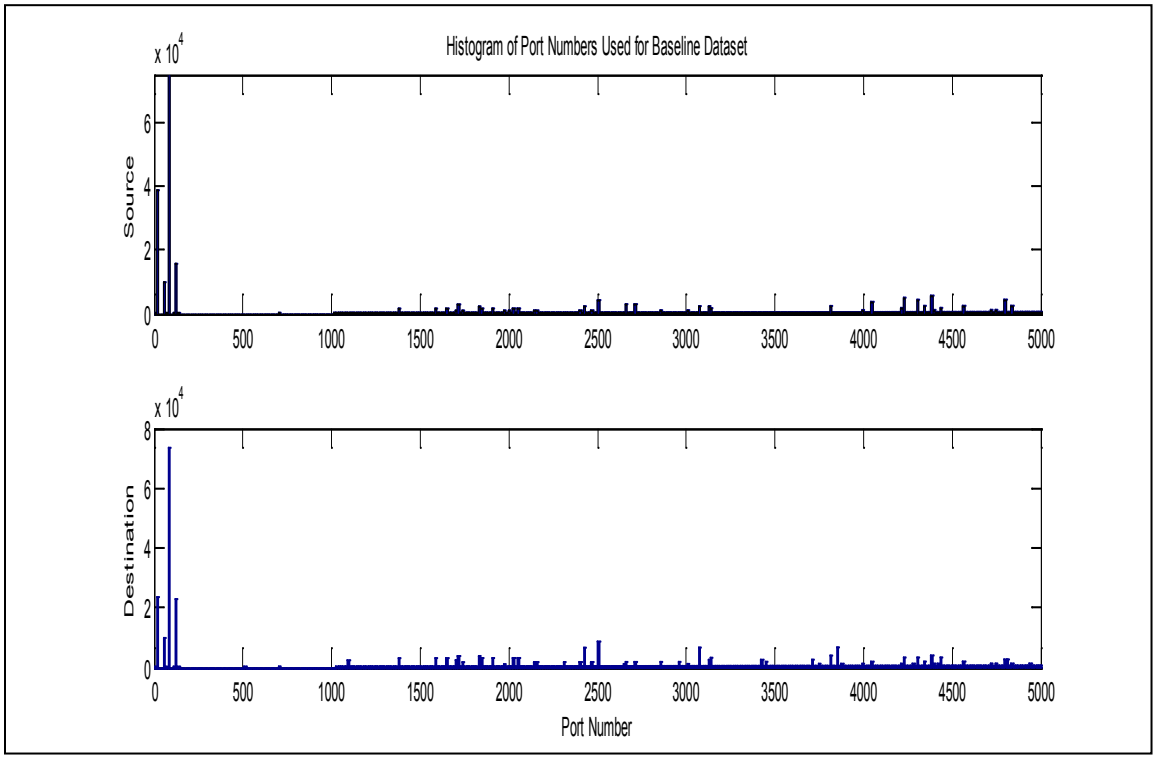

Figure 4. Histogram of Ports, Baseline Dataset

The top histogram is the source ports with the bottom being the destination port. While there is a definitive disparity toward the lower ports, specifically port 80, there are still a large distribution of other ports used.

3.4. System Parameters and Factors

As seen in Table 2, each line of data in the datasets contains thirteen fields about each packet that passed through the collector. Different attacks have different signatures indicating an attack. Different fields are needed for examination to determine if an attack is occurring or has occurred. Not every element for each data packet is needed for analysis depending on attack type. Since the attack types for each dataset are known, this

experiment only uses the fields that are primarily associated with detection of these attacks, namely the 'win,' 'buf,' and 'ulen' fields.

These fields are used individually for the matrix population and SVD compression. Additionally the 'win' and 'ulen' values were combined and populated in to a single matrix. This combined matrix is used to represent the overall volume of data entering the network. The 'buf' field was not needed because every packet that contained a 'win' value also contained a 'buf' value, so while the value would change the matrix placement would not.

Table 2. Dataset Elements

Data Element Field Name	**Description**
Time	Converted to floating point seconds measured with 0.00001 precision
Scraddr	Final octet of source IP address
Srcport	Port address from the source address
Destaddr	Final octet of destination IP address
Destport	Port address for the destination address
Flag	Special designation for packet can contain syn, fin, push, rst, U, X, or XPE
Seq1	Data sequence number of packet for nonUDP packets
Seq2	Data sequence number of the data expect in return packet
Ack	Data sequence number of the next expected packet expected from the other direction on this connection
Win	Number of bytes of receive buffer space available from the other direction on this connection
Buf	Number of bytes of receive buffer space available on this connection
Ulen	Length in bytes of UDP packet
Op	Optional information about the packet such as DF – do not fragment

Time Frame: One of the goals for network monitoring tools is detection as close to near real time as possible. It was critical to choose data with a time stamp to simulate data used by a monitoring tool. As previously mentioned the KDD'99 dataset, did not contain a time stamp. Time intervals are chosen to see if there is any statistical difference in selected network traffic fields of the different datasets for varying intervals, as some networks can exhibit periods of greater traffic volume in shorter time intervals. However, too small of a time frame could yield too little information for analysis and would create a need to combine multiple matrices. Initial analysis containing the average number of packets in each time frame for all the datasets are shown in Table 3.

Table 3. Average Packets Over Time

Dataset	**0.25 sec**	**0.5 sec**	**1.0 sec**	**2.0 sec**
Baseline	95.516	191.032	382.0650	764.13
Network 1	130.234	260.469	520.9395	1041.879
Network 2	111.333	222.666	445.3327	890.6654
Network 3	111.387	222.774	445.5486	891.0972
Network 4	130.910	261.821	523.6413	1047.283

Based on this initial analysis, the experiment alters the time frame of packets analyzed through a range of 0.25, 0.5, 1.0, and 2.0 second intervals.

Matrix size: Maintaining a constant matrix size solidifies value placement in accordance with the matrix population method described in Section 3.1. Additionally, given system restraints, a constant matrix size simplifies the calculation and implementation of SVD compression.

Computer networks ports have 2^{16} or 65,536 ports. Not all ports are used at all times. Most common or critical system functions have very low port numbers below 1024, and are considered "reserved." The higher ports are open and typically used by

different software applications. Since the lower numbered ports are more common, having a standard size for the matrix allows for determination of what service or what type of application is used. If a value or quantity of values is present in a specific row or column, this indicates a specific port or process was used during that time frame. Based on average number of packets for the longest time interval for all the datasets, the matrix size chosen for this experiment is 200x200 cells, giving over 40,000 different positions for a two second interval.

Source and destination ports: The IP addresses were masked and always contained the value "2" in either source or destination addresses. This can be observed in Appendix A. The lack of diversity in IP addresses disallowed their use for matrix population However, the source and destination ports showed much more variation for each packet throughout the duration of each dataset, as shown in Figure 4. Because the matrix size has been set to 200x200, there is potential for values from different ports that are close to one another to combine in the same cell. The use of small time frames decreases this occurrence with the Maximum average ports per matrix of 104 ports used between both source and destination, as shown in Table 4 below.

Table 4. Dataset Port Analysis

	Total # of ports used	Avg # of ports per Matrix at 2.0 sec
Baseline	4245	70.1319 Std Dev 8.9015
Network1	4269	103.9238 11.5706
Network 2	4197	99.0850 11.6786
Network 3	4281	99.6713 10.7611
Network 4	4269	104.2897 11.6447

This allowed the use of port as a method of positioning the individual packet element values in the matrix fields. Additionally, having the size of the matrix fixed allowed for consistent placement of packet element data based on port number and direction of packet flow, either outbound or inbound. Given time frames selected for the experiment, the number of ports used per matrix should allow for a sparsely populated matrix.

Buffer Size: The 'win,' 'buf,' and 'ulen' fields in the datasets determined the amount of data transferred in the individual packets. These values are used and potentially combined in the created matrix depending on the flow of the traffic over the selected time interval.

Compression size: Singular Value Decomposition compression operates on the basis of eliminating the singular values that are least important to the original matrix. Therefore in order to compress a matrix by 50%, one needs to eliminate the lowest 50% of the singular values. As stated before, the goal is to reduce the matrix as much as possible without altering the individual values from the original matrix. The compression ratio starts at 50% and adjusts to compress more or less until the highest level of compression is achieved without any of the values of the original matrix being altered.

3.5. Performance Metrics

The metrics taken from this experiment are comparisons between the compressed matrix to the original matrix, and the change in values before and after the SVD compression has been applied. These values varied based on time frame selected for the individual experiments. If there is a difference between individual fields, the difference

is recorded and compared to the value of the original uncompressed matrix. The values taken are:

Maximum difference value: This is the Maximum value of the difference between the entire set of non-compressed and compressed matrices. This value indicated how much the values from the original matrix have changed due to loss of data from the SVD compression. This value is not allowed to be greater than one for the Maximum compression matrix sets.

Maximum compression: This value is the amount of removed data from the original decomposition matrices; e.g 80% indicates that the decomposition matrices only retain 20% of the original data. This is the value at which further compression results in the individual values occupying the matrix begin to vary by greater than one from the original value. Determination of this value requires many simulations at different levels of compression.

Mean number of Eigenvalues: The eigenvalues, when ordered, determine which values carry the most weight. Eliminating the eigenvalues with the least value assist in determining how much compression will be needed in determining maximum compression.

Mean number of Eigenvalues greater than one: This helped determine the Maximum compression value. The mean number of these significant eigenvalues provided insight into how much of the matrix to reduced during compression. Initial analysis of matrices indicates that the eigenvalues for each matrix are composed of values that are either much greater than one or much less than one.

Mean number of values per matrix: This measurement helps determine how sparse the matrix is. If this value is too low then compression is a futile effort and no further analysis of this matrix is necessary. Since the matrices are fixed in size, each matrix is anticipated to be mostly empty or sparse. A simple count of non-zero values in the original matrix accounts for how many should be in the compressed matrix.

Mean number of combined values in the decomposition: This value examines the number of non-zero values for each of the three reduced matrices after compression. This is the minimum number of values needed to recalculate the original matrix. This value does not count the zero values because of the use of the use of a matrix format.

3.6. Experimental Design

The experiment is partial factorial; each network dataset is divided and parsed along the four time intervals with the three selected data fields over the entire twenty minute duration. The datasets provided represent the same network with a baseline and four attack datasets. The purpose of the experiment is not to distort the data, but to allow for the data to be used by a network monitoring tool or administrator. The fact that the network is under attack does not interfere with the use the data. This allows for all datasets to be compared relative to one another in evaluating the compression technique. The sample sizes for each dataset varied based on selected time frame. The fewest number of samples in all the datasets is in the baseline 2.0 second analyses, which yielded the smallest sample size of 470 matrices throughout the entire experiment.

Each of the four time frame samples is reduced twice. The first is to 50% and the second is referred to as Maximum compression. Maximum compression is defined as the

lowest compression level SVD can achieve without altering any of the individual values by more than one. The eigenvalues of the original matrix are analyzed to aide in the determination of the Maximum compression value. Both quantity and value of the eigenvalues are observed to determine how much reduction should be implemented. Simulations are run until the Maximum compression is determined by increasing or decreasing the amount of values eliminated by SVD compression. Once the Maximum compression value is determined then the quantity of values in the decompositions are compared to the number of original data values.

3.7. Evaluation Technique

To evaluate this experiment multiple measurements are taken of the matrices and matrix values before and after SVD compression has been applied. To validate that the compression algorithm does not eliminate too much data, the values of the individual matrices cells are compared from before and after SVD compression in two ways. Based on matrix size and number of ports used per time frame, most cell values of the original matrices are zero. It is expected that these zero values are the most likely to change due to SVD compression. Therefore, in order to eliminate insignificant data created through the SVD compression algorithm, only the values greater than one are counted. This quantity of values should be equal to the number of values in the original matrix; the first comparison. Secondly, the individual values of the cells counted are compared to the original matrix. If these individual values have a change of less than one, then the compression is considered successful.

The Maximum compression is determined, as the maximum amount of reduction performed to the decomposition matrices without the resulting calculated matrix values being altered by more than a value of one. Once the Maximum compression has been determined, the mean number of values of each reduced decomposition matrix counted. Only the values greater than one are counted, as all other values are considered altered zero values from the original matrix by the SVD compression process. The total quantity of values in the decomposition matrices is then compared to the total number of values needed to create the original matrix.

These quantities and values are gathered on all five datasets and at each time frame sample. This allowed for a large sample set of how the SVD compression algorithm performs given the varying nature of the network dataset values. A two sample t-test comparing: the samples of values used to create the original matrix and the quantity of values counted from the decomposition for each matrix, is conducted to determine if these sets of data are statistically different from each other.

3.8. Summary

A known and tested network traffic data set, IES, is used and parsed into NxN matrixes based on individual data fields that are network traffic measurements. The matrices are then compressed using singular value decomposition. The compression level is equivalent to Maximum amount of compression where the matrix values do not vary by more than one. The individual values of the compressed matrices are compared to the original to determine if any significant variance has occurred. The total number of values created from the decomposition after compression are measured to compare

against the total number of values that would need to be sent to equal the amount of data put into each matrix. The IES dataset contains five separate network captures representing first a baseline network and four captures featuring a single network attack in each network. The datasets parsed the packet capture data using four different time intervals of 0.25, 0.5, 1.0 and 2.0 seconds for the entire dataset.

The resulting values in the compressed matrices are compared to the values of the original matrices. In addition, after compressing the matrices, the quantity of values in the resulting decomposition matrices is compared to the number values from original data needed to recreate the data from the original matrix.

IV. Analysis and Results

4.1. Results of Simulation Scenarios

4.1.1. Baseline dataset analysis:

When looking at the data for the 'win' field, it was observed that the number of the eigenvalues generated from the original matrices tend to show that each matrix generates a significant number of eigenvalues, only about half of them are of significant value at or greater than on the value of one. Compressing the matrices to 50% allowed keeping all of the eigenvalues for each time frame and the resulting compressed matrix contained near identical data with the largest difference in value from any matrix being significantly less the one byte value. The number of values needed for the decomposition matrices in 0.25, 0.5, and 1.0 second time frame was not less than the total number values from the original data, as seen in first row of Table 5. However the 2.0 sec time frame for Maximum compression did decrease the overall number of values from the original data in Table 6.

Table 5. Uncompressed matrix using just 'win' field - Baseline

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	382.065	764.13	1528.2	3056.5
Mean # values per matrix	35.5243 Std 7.9278	48.6955 Std 9.6032	66.5123 Std 11.3596	92.2234 Std 14.4679
Mean # of ports used per matrix	32.2775 Std 6.1890	41.1874 Std 6.8655	53.6499 Std 7.3363	70.1319 Std 8.9015
Mean # of Eigenvalues per matrix	12.4809 Std 4.1601	16.3577 Std 6.7512	24.9797 Std 11.1454	36.4043 Std 13.6312
Mean # of Eigenvalues < 1 per matrix	9.7653 Std 2.7797	11.6821 Std 3.0334	13.7780 Std 3.2387	16.4766 Std 3.2327
Maximum % compression	87	85	83	80

Table 6. Compressed matrices using 'win' field - Baseline

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of combined values of decomposed matrices at 50%	15291	19640	24652	29581
Mean # values per matrix greater than 1 at 50% compression	35.5243 7.9278	48.6955 9.6032	66.5123 11.3596	92.2234 14.4679
Maximum difference in values from original matrix	4.556×10^{-13}	9.017×10^{-13}	2.630×10^{-12}	1.148×10^{-11}
Mean # of combined values of decomposed matrices at Maximum	560.655	963.4581	1599.4	2875.8
Mean # values per matrix greater than 1 at Maximum compression	35.5243 7.9278	48.6955 9.6032	66.5123 11.3596	92.2234 14.4679
Maximum difference in values from original matrix	4.556×10^{-13}	9.017×10^{-13}	2.630×10^{-12}	1.147×10^{-12}

The 'buf' field data yielded similar results in Table 7 and Table 8. There were approximately 20 entries per matrix making these matrices sparse than the 'win' field. There were also very few eigenvalues generated by each matrix. The 50% compression allowed for all eigenvalues to be kept, and that all the values of the matrix did not change by more than a value of 0.004. For Maximum compression, a closer look at the eigenvalues for each time frame provides evidence that even though the eigenvalues increased as the amount of data per matrix increased, the amount of eigenvalues greater than one did not increase at the same rate. The low amount of significant eigenvalues allowed for a Maximum compression for each matrix set greater than 50%. The Maximum compression sets yielded a reduction in number of values from the original data in every time frame by 76.8%, 66.7%, 66.2%, and 68.4% respectively.

Table 7. Uncompressed matrix using just 'buf' field - Baseline

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	382.065	764.13	1528.2	3056.5
Mean # values per matrix	19.3217 4.5799	28.5618 5.7907	41.2085 7.3254	61.2085 10.4500
Mean # of ports used per matrix	32.2775 6.1890	41.1874 6.8655	53.6499 7.3363	70.1319 8.9015
Mean # of Eigenvalues per matrix	3.6921 2.1660	6.5921 2.4747	9.4771 3.2241	13.0766 6.1341
Mean # of Eigenvalues < 1 per matrix	3.2452 1.888	5.3228 1.9468	6.9146 1.9040	8.1149 1.9094
Maximum % compression	95	93	91	90

Table 8**.** Compressed matrices using just 'buf' field - Baseline

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	1327.9	2914.3	5136.1	8025.8
Mean # values < 1 at 50% compression per matrix	19.0019 4.5417	28.2916 5.7889	41.6830 7.3562	60.9404 10.4716
Maximum difference in values from original matrix	0.0014	0.0016	0.0032	0.0043
Mean # of values from SVD at Maximum	88.6709	254.7293	516.1217	966.3106
Mean # values < 1 at Maximum SVD per matrix	14.9690 3.4717	26.3061 5.1106	41.0438 7.1319	60.2213 10.2889
Maximum difference in values from original matrix	0.0901	0.0279	0.0619	0.0340

Due to small amount of UDP packets in Table 9 for this dataset there is no need for further analysis of only UDP packet.

Table 9. Uncompressed matrix using just 'ulen' field - Baseline

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values per matrix	0.8520 0.5123	0.9753 0.6004	1.686 0.7813	1.3957 1.2098

The final set of matrices combined the fields 'win' and 'ulen' fields, Table 10 and Table 11. This matrix performed similarly to the singular 'win' field in regards to Maximum compression and overall performance in values produced form the decomposition matrices. The Maximum compression did value did decrease slightly compared to the singular 'win' field. This is due to the low volume of 'ulen' packets passing through the network over the course of the dataset. Both 50% and Maximum compression for this set of matrices yielded no significant reduction in values.

Table 10. Uncompressed matrix using combined 'win' and 'ulen' fields - Baseline

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	477.5813	955.1625	1910.325	3820.65
Mean # values per matrix	36.3763 7.9285	49.6708 9.5888	67.6809 11.3513	93.6191 14.48.35
Mean # of ports used per matrix	32.2775 6.1890	41.1874 6.8655	53.6499 7.3363	70.1319 Std 8.9015
Mean # of Eigenvalues per matrix	12.4806 4.1557	16.3711 Std 6.7560	24.9936 Std 11.1426	36.4553 Std 13.5945
Mean # of Eigenvalues < 1 per matrix	9.7996 2.7794	11.6907 3.0350	13.7930 3.2440	16.5021 3.2440
Maximum % compression	87	85	83	80

Table 11. Compressed matrices using combined 'win' and 'ulen' fields - Baseline

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	3392.4	5149.8	7435.6	10895
Mean # values < 1 at 50% compression per matrix	36.3763 7.9285	49.6708 9.5888	67.6809 11.3513	93.6191 14.4835
Maximum difference in values from original matrix	4.556×10^{-13}	9.017×10^{-13}	2.630×10^{-12}	1.148×10^{-11}
Mean # of values from SVD at Maximum	567.6340	986.3695	1656.6	3021.3
Mean # values < 1 at Maximum SVD per matrix	36.3763 7.9285	49.6708 9.5888	67.6809 11.3513	93.6191 14.4835
Maximum difference in values from original matrix	4.556×10^{-13}	9.017×10^{-12}	2.630×10^{-12}	1.148×10^{-11}

Overall Summary:

The SVD compression algorithm appears to function well at levels much greater than 50% reduction of decomposition size. Eigenvalues for all matrices and all fields appear to fall into two categories: much greater than one or much less than one. Though there are a significant number of eigenvalues that are much less than one, most of these can be removed without affecting the values in the original matrix. SVD compression seems to create a 'noise floor' in terms of values in the recombined compressed matrix. The values are changed are not changed much as the tables indicate.

4.1.2. Network 1 dataset analysis:

For the Network 1 dataset 'win' field, there is significantly more data per matrix as seen in the first row of the table below. The 50% compression allowed for all original eigenvalues to be kept for all time frames, and shows no significant change in values as seen in Table 12 and Table 13. However, the result of the compression using the Maximum compression size remains the same with no significant difference in value. It was observed that while the eigenvalues increased significantly between the time frames, the number of eigenvalues greater than one did not increase at the same rate. The number of data values for each of the compressed matrix sets was not less than the original number of values needed to create the matrix.

Table 12. Uncompressed matrix using just 'win' field value – Network 1

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	520.9395	1041.879	2083.758	4167.516
Mean # values per matrix	56.3949 10.8051	79.8048 13.2184	112.0565 14.6028	152.5000 18.6947
Mean # of ports used per matrix	45.9411 8.0200	59.9057 8.8850	79.7060 9.3766	103.9238 11.5706
Mean # of Eigenvalues per matrix	21.8033 8.3342	36.3609 10.9554	50.0556 15.1289	69.8576 19.6401
Mean # of Eigenvalues < 1 per matrix	14.3859 3.5200	18.2286 3.8940	21.5540 4.1119	24.8874 4.4470
Maximum % compression	82	78	74.5	70.5

Table 13. Compressed matrices using 'win' field value - Network 1

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	6550.1	9932.6	13995	17980
Mean # values < 1 at 50% compression per matrix	56.3949 10.8051	79.8048 13.2184	112.0565 14.6028	152.5000 18.6947
Maximum difference in values from original matrix	6.005×10^{-13}	1.123×10^{-12}	5.533×10^{-12}	2.211×10^{-11}
Mean # of values from SVD at Maximum	1523.7	2958.9	5415.2	8869.5
Mean # values < 1 at Maximum SVD per matrix	56.3949 10.8051	79.8048 13.2184	112.0565 14.6028	152.5000 18.6947
Maximum difference in values from original matrix	6.005×10^{-13}	1.123×10^{-12}	5.533×10^{-12}	2.211×10^{-11}

The 'buf' field also increased in number of values per matrix when compared to the baseline dataset. A closer evaluation of these eigenvalues shows they follow the same trend either belonging into one of two categories much greater than one or much less than one. All values maintained their original values without significant change. The number of data values for each of the compressed matrix sets was not less than the original number of values needed to create the matrix shown in the difference between Table 14 and Table 15.

Table 14. Uncompressed matrix using just 'buf' field value - Network 1

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	520.9395	1041.879	2083.758	4167.516
Mean # values per matrix	41.1454 7.5908	62.7626 9.9912	94.1736 11.9776	135.7401 15.8890
Mean # of ports used per matrix	45.9411 8.0200	59.9057 8.8850	79.7060 9.3766	103.9238 11.5706
Mean # of Eigenvalues per matrix	13.6840 4.2439	22.0096 7.7416	37.9435 10.9366	58.1010 15.8949
Mean # of Eigenvalues < 1 per matrix	10.8359 3.0269	14.9616 3.3441	19.1968 3.5823	23.3709 4.2114
Maximum % compression	86	82	77	72

Table 15. Compressed matrices using just 'buf' field value - Network 1

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	4860.1	8155.8	12254	16561
Mean # values < 1 at 50% compression per matrix	40.5502 7.5065	62.0709 9.8979	93.3721 11.8970	134.9719 15.8222
Maximum difference in values from original matrix	6.312×10^{-4}	0.0013	0.0025	0.0050
Mean # of values from SVD at Maximum	813.1096	1866.7	3931.4	7241.5
Mean # values < 1 at Maximum SVD per matrix	40.5369 7.4928	62.0697 9.8968	93.3721 11.8970	134.9719 15.8222
Maximum difference in values from original matrix	6.312×10^{-4}	0.0013	0.0025	0.0050

Due to small amount of UDP packets for this dataset, in Table 16, there is no need for further analysis of only UDP packets.

Table 16. Uncompressed matrix using just 'ulen' field - Network 1

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values per matrix	0.0762 0.2910	0.1519 0.4260	0.3023 0.6156	0.5977 0.9699

The combined field matrix for Network 1, Table 17 and Table 18, behaved similar to the 'win' matrix set. The increased average number of packets per matrix did not

significantly affect this matrix set, as the values from the 'ulen' field did not significantly alter the values after either 50% or Maximum compression. Nor did the addition of the 'ulen' data significantly alter the total number of eigenvalues per matrix. The number of data values for each of the compressed matrix sets was not less than the original number of values needed to create the matrix.

Table 17. Uncompressed matrix using combined 'win' and 'ulen' fields - Network 1

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	651.1744	1302.349	2604.698	5209.395
Mean # values per matrix	56.4681 10.8121	79.9433 13.2327	112.3115 14.6207	152.8957 18.6957
Mean # of ports used per matrix	45.9411 8.0200	59.9057 8.8850	79.7060 9.3766	103.9238 11.5706
Mean # of Eigenvalues per matrix	21.8146 8.3350	36.3880 10.9542	50.0482 15.1424	70.0033 19.5225
Mean # of Eigenvalues < 1 per matrix	14.5556 3.5205	18.2520 3.8948	21.5706 4.1106	24.9040 4.4466
Maximum % compression	82	78	74	71

Table 18. Compressed matrix using combined 'win' and 'ulen' fields - Network 1

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	6550.4	9933	13996	17982
Mean # values < 1 at 50% compression per matrix	56.3983 10.8060	79.8114 13.2189	112.0698 14.6009	152.5232 18.6857
Maximum difference in values from original matrix	6.005×10^{-13}	1.123×10^{-12}	5.533×10^{-12}	2.211×10^{-11}
Mean # of values from SVD at Maximum	1523.8	2959.3	5415.8	8872.2
Mean # values < 1 at Maximum SVD per matrix	56.3983 10.8060	79.8114 13.2189	112.0698 14.6009	152.5232 18.6857
Maximum difference in values from original matrix	6.005×10^{-13}	1.123×10^{-12}	5.533×10^{-12}	2.211×10^{-11}

Overall Summary:

Due to the significant increase in the number of packets per time period, there is a large amount of data per matrix. As observed in the Baseline dataset, there is a growing trend of an increase in number of values per matrix leads to a greater increase of eigenvalues whose value is much less than one and only a slight increase in the number of eigenvalues greater than one. Additionally all eigenvalues have continued to be much greater than one or much less than one. None of these compression sets contained fewer values than the original values needed to create the matrix.

4.1.3. Network 2 dataset analysis:

The 'win' field for Network 2 dataset, in Table 19 and Table 20, showed there were significantly more values per matrix than the Baseline, though not as much as in Network 1. In these experiments, the compression continues to show no significant change in number of values created from the decomposition matrices. The number of eigenvalues continues to increase as the number of data increases. The minimum value for each time frame shows a large minimum nonzero value.

Table 19. Uncompressed matrix using just 'win' field - Network 2

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	445.3327	890.6654	1781.331	3562.662
Mean # values per matrix	52.2233 10.7007	74.7407 13.5907	105.9722 15.5858	143.5213 19.3885
Mean # of ports used per matrix	43.1967 7.9450	56.9865 9.0869	76.4352 9.7159	99.0850 11.6786
Mean # of Eigenvalues per matrix	20.5946 7.5184	34.2509 11.0529	47.4167 14.6010	64.0573 20.2937
Mean # of Eigenvalues < 1 per matrix	14.5556 3.5058	18.2520 3.8688	21.5706 3.8909	24.9040 3.9966
Maximum % compression	82	78	74	71

Table 20. Compressed matrices using 'win' field value - Network 2

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	5120	7917.1	11697	15783
Mean # values < 1 at 50% compression per matrix	52.2233 10.7007	74.7407 13.5907	105.9722 15.5858	143.5213 19.3885
Maximum difference in values from original matrix	8.405×10^{-13}	2.076×10^{-12}	5.188×10^{-12}	2.066×10^{-11}
Mean # of values from SVD at Maximum	1163.4	2274.5	4327.1	7389.6
Mean # values < 1 at Maximum SVD per matrix	52.2216 10.6970	74.7407 13.5907	105.9722 15.5858	143.5213 19.3885
Maximum difference in values from original matrix	8.405×10^{-13}	2.076×10^{-12}	5.188×10^{-12}	2.066×10^{-11}

The matrices for 'buf' field in Network 2, Table 21 and Table 22, initially appeared similar to the values in of Table 14 and Table 15, where the matrices contained few eigenvalues. While this allowed for greater Maximum compression, it did increase significantly the difference in value after the compression. The 50% compression did not produce fewer values than the original data. The Maximum compression yielded a reduction in total number of values for all the time frames. The number of values per matrix decreased on average by 63.7%, 42.6%, 29%, and 39.5% respectively for each time frame sample.

Table 21. Uncompressed matrix using just 'buf' field value - Network 2

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	445.3327	890.6654	1781.331	3562.662
Mean # values per matrix	28.9370 6.8600	44.4777 8.9827	68.3602 9.7428	96.2089 12.3035
Mean # of ports used per matrix	43.1967 7.9450	56.9865 9.0869	76.4352 9.7159	99.0850 11.6786
Mean # of Eigenvalues per matrix	5.4533 2.8362	9.7658 3.6147	15.6093 5.5724	24.8872 10.4096
Mean # of Eigenvalues < 1 per matrix	4.4905 2.2408	7.4905 2.2912	9.8528 1.9544	11.3826 1.9653
Maximum % compression	93	90	88	87

Table 22. Compressed matrix using just 'buf' field - Network 2

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	2080.1	4604.6	8380.7	11960
Mean # values < 1 at 50% compression per matrix	28.9370 6.8600	44.4777 8.9827	68.3602 9.7428	96.2089 12.3035
Maximum difference in values from original matrix	0	0	0	0
Mean # of values from SVD at Maximum	161.7598	511.6243	1264.7	2154.8
Mean # values < 1 at Maximum SVD per matrix	21.5824 4.7205	40.9181 7.7695	67.0861 9.2960	94.6728 11.9738
Maximum difference in values from original matrix	0.1203	0.0889	0.0315	0.0628

Due to small amount of UDP packets, as seen in Table 23, for this dataset there is no need for further analysis of only UDP packets.

Table 23. Uncompressed matrix using just 'ulen' field - Network 2

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values per matrix	0.0585 0.2436	0.1145 0.3370	0.2222 0.4481	0.3494 0.5322

The results, shown in Table 24 and Table 25, contain the combined 'win' and 'ulen' packet fields. These findings were similar results to Network 1 shown in Table 17

and Table 18. The data compressed to the 50% level with no significant change in values of the matrix. Looking at the eigenvalues of this series, the values continue to be much greater than one and much less than one. Maximum compression does not produce fewer values from its decomposition matrices on average for any time frame.

Table 24. Uncompressed matrix using combined 'win' and 'ulen' fields - Network 2

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	556.6659	1113.332	2226.664	4453.327
Mean # values per matrix	52.2818 10.7033	74.8552 13.5946	106.1944 15.5875	143.8706 19.3821
Mean # of ports used per matrix	43.1967 7.9450	56.9865 9.0869	76.4352 9.7159	99.0850 11.6786
Mean # of Eigenvalues per matrix	20.6026 7.5211	34.2924 11.0481	47.3787 14.6305	63.9335 20.4051
Mean # of Eigenvalues < 1 per matrix	14.5556 3.5063	18.2700 3.8664	21.6593 3.8901	25.1867 3.9931
Maximum % compression	82	78	74	71

Table 25. Compressed matrix using combined 'win' and 'ulen' fields - Network 2

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	5121.2	7922.3	11714	15830
Mean # values < 1 at 50% compression per matrix	52.2818 10.7033	74.8552 13.5946	106.1944 15.5875	143.8706 19.3821
Maximum difference in values from original matrix	8.405×10^{-13}	2.076×10^{-12}	5.188×10^{-12}	2.066×10^{-11}
Mean # of values from SVD at Maximum	1164.7	2280.3	4348.7	7442.3
Mean # values < 1 at Maximum SVD per matrix	52.2818 10.7033	74.8552 13.5946	106.1944 15.5875	143.8706 19.3821
Maximum difference in values from original matrix	8.405×10^{-13}	2.076×10^{-12}	5.188×10^{-12}	2.066×10^{-11}

Overall Summary:

Continued trends from the first two data sets persist: the number of significant eigenvalues increase at a much slower rate than the overall number of eigenvalues in each

matrix. The increased time frame allowed for more data per matrix increasing the Maximum compression. The number of values in the Maximum compressed matrix set of the 'buf' field was consistently less than the number of original data values. The 'buf' field was the only field to successfully reduce the overall number of values. Both the other matrices increased the number of values from the original dataset.

4.1.4. Network 3 dataset analysis:

The results for matrix set using the 'win' field of Network 3,seen in Table 26 and Table 27, yielded a greater amount of compression per time frame than in Network 2. When comparing the two networks, the number of values per matrix is approximately the same over the entire dataset, as shown in the first row of Table 19 and Table 26. The behavior in this experiment is also similar. However, the greater compression percentage for each time frame does not reduce the overall number of values from the original data.

Table 26. Uncompressed matrix using just 'win' field - Network 3

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	445.5486	891.0972	1782.194	3564.389
Mean # values per matrix	51.0406 10.6842	73.1738 12.9950	105.3649 14.5850	144.3794 17.9455
Mean # of ports used per matrix	41.8498 7.6831	55.6176 8.4786	75.7746 8.9996	99.6713 10.7611
Mean # of Eigenvalues per matrix	19.0350 7.1264	31.8319 11.0038	45.8070 14.0745	63.1381 19.7636
Mean # of Eigenvalues < 1 per matrix	13.1684 3.2296	16.1539 3.727	19.3491 3.4322	22.7133 3.4073
Maximum % compression	84	80	77	74

Table 27. Compressed matrix using 'win' field - Network 3

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	5552.9	8714	13017	16994
Mean # values < 1 at 50% compression per matrix	51.0406 10.6841	73.1738 12.9950	105.3649 14.5850	144.3794 17.9455
Maximum difference in values from original matrix	2.228×10^{-4}	1.335×10^{-12}	6.964×10^{-12}	1.728×10^{-11}
Mean # of values from SVD at Maximum	1292.1	2444.9	4702.7	7539.4
Mean # values < 1 at Maximum SVD per matrix	51.0406 10.6841	73.1738 12.9950	105.3649 14.5850	144.3794 17.9455
Maximum difference in values from original matrix	2.228×10^{-4}	1.335×10^{-12}	6.964×10^{-12}	1.728×10^{-11}

The 'buf' field results are shown in Table 28 and Table 29. The Maximum compressions for this field yielded fewer values in the decomposition matrices for each time frame than the number of values used in the original dataset. The percentage reduced varied from each time frame, with the most reduction occurring at 0.25sec with 41.6% in number of values on average through the entire dataset. The other time frames reduced the number of values by 14.8%, 5.9%, and 25% respectively. This reduction can be attributed to the mean number of values per matrix for the 0.25 seconds time frame being almost 35% less than the 0.5 seconds time frame on average, see Table 28.

Table 28. Uncompressed matrix using just 'buf' field - Network 3

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	445.5486	891.0972	1782.194	3564.389
Mean # values per matrix	28.8643 6.8975	44.6859 8.6656	69.3623 9.6298	98.4790 12.5320
Mean # of ports used per matrix	41.8498 7.6831	55.6176 8.4786	75.7746 8.9996	99.6713 10.7611
Mean # of Eigenvalues per matrix	6.2373 2.8228	10.9237 3.6756	16.8816 6.0747	28.3059 11.0850
Mean # of Eigenvalues < 1 per matrix	5.1277 2.1989	8.0922 2.0589	10.2842 1.8043	11.5490 1.8743
Maximum % compression	93	90	88	87

Table 29. Compressed matrix using just 'buf' field - Network 3

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	2525.5	5445.7	9454.4	13172
Mean # values < 1 at 50% compression per matrix	27.7975 6.7816	43.8633 19.1279	68.6456 9.5635	97.8916 12.5654
Maximum difference in values from original matrix	0.0018	0.0022	0.0044	0.0070
Mean # of values from SVD at Maximum	260.178	759.0159	1677.6	2673.7
Mean # values < 1 at Maximum SVD per matrix	25.1662 5.7424	43.2942 8.3394	68.3482 9.4386	97.7832 12.5444
Maximum difference in values from original matrix	0.0111	0.0106	0.0070	0.0105

Due to small amount of UDP packets, seen in Table 30, for this dataset there is no need for further analysis of only UDP packets

Table 30. Uncompressed matrix using just 'ulen' field - Network 3

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values per matrix	0.0555 0.2394	0.1054 0.3240	0.1825 0.4170	0.2640 0.4972

Table 31 and Table 32 represent the data gathered from the combined 'win' and 'ulen' fields. This matrix set for Network 3 behaved similar to the singular 'win' field

matrix. The number of overall eigenvalues dramatically increased with the addition of the 'ulen' data. The number of significant eigenvalues did not altered significantly, which is illustrated in the average Maximum compression remaining similar to the values of the 'win' field illustrated in Table 26. The compressions for this matrix set did not produce a lower number of values than the original data.

Table 31. Uncompressed matrix using combined 'win' and 'ulen' fields - Network 3

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	556.9358	1113.872	2227.743	4455.486
Mean # values per matrix	51.0960 10.6965	73.2792 13.0182	105.5474 14.6358	144.6434 18.0365
Mean # of ports used per matrix	41.8498 7.6831	55.6176 8.4786	75.7746 8.9996	99.6713 10.7611
Mean # of Eigenvalues per matrix	51.0954 10.6961	73.2792 13.0182	105.5474 14.6358	144.6434 18.0365
Mean # of Eigenvalues < 1 per matrix	13.1684 3.2328	16.1539 3.3787	19.3491 3.4443	22.7133 3.4348
Maximum % compression	84	80	77	74

Table 32. Compressed matrix using combined 'win' and 'ulen' fields - Network 3

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	5554.2	8724.6	13046	17037
Mean # values < 1 at 50% compression per matrix	51.0948 10.6951	73.2792 13.1082	105.5474 14.6358	144.6434 18.0365
Maximum difference in values from original matrix	2.223×10^{-4}	1.335×10^{-12}	6.964×10^{-12}	1.728×10^{-11}
Mean # of values from SVD at Maximum	1293.4	2455.8	4733.5	7585.1
Mean # values < 1 at Maximum SVD per matrix	51.0945 10.6951	73.2792 13.1082	105.5439 14.6358	144.6434 18.0365
Maximum difference in values from original matrix	2.223×10^{-4}	1.335×10^{-12}	6.964×10^{-12}	1.728×10^{-11}

Overall Summary:

Trends established in previous network datasets continue to hold. The number of eigenvalues continues to grow rapidly as the number of values per matrix increase. The number of significant eigenvalues increases at a much slower rate, indicating that there is an increase in overall quantity of values per matrix. However, most of the additional data is being combined in the matrix. Only the 'buf' field compressions yielded a lower total quantity of values than the original data. When comparing 'buf' to the other fields test in this dataset, the most notable difference is the number of values per matrix. The 'buf' field contained significantly less values per matrix than the others per time frame.

4.1.5. Network 4 dataset analysis:

The Network 4 analysis yielded the highest average number of packets per time frame for all the datasets. The 'win' field failed to produce decompositions with fewer values than the original data for matrix creation, shown in Table 34. The Maximum compression was found to be less than in the previous datasets, shown in Table 33. This aligns with the trend of number of values per matrix. Network 4 also contained the highest number of values per matrix indicating a large amount of traffic not being combined over common ports.

Table 33. Uncompressed matrix using just 'win' field - Network 4

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	523.6413	1047.283	2094.565	4189.13
Mean # values per matrix	56.8561 10.8863	80.3557 13.3984	112.8846 14.7869	153.4868 18.8882
Mean # of ports used per matrix	46.1353 8.0303	60.1134 8.9307	80.0249 9.4363	104.2897 11.6447
Mean # of Eigenvalues per matrix	22.1592 8.4628	36.3153 11.3776	50.2874 15.3333	70.1308 19.9094
Mean # of Eigenvalues < 1 per matrix	14.6218 3.5065	18.2794 3.8867	21.5706 4.1150	24.8907 4.4470
Maximum % compression	82	78	74	71

Table 34. Compressed matrix using 'win' field - Network 4

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	6656.2	10001	14078	18089
Mean # values < 1 at 50% compression per matrix	56.8561 10.8863	80.3557 13.3984	112.8846 14.7869	153.4868 18.8882
Maximum difference in values from original matrix	5.977×10^{-13}	1.157×10^{-12}	5.506×10^{-12}	2.375×10^{-11}
Mean # of values from SVD at Maximum	1547.6	2973.5	5454.7	8927
Mean # values < 1 at Maximum SVD per matrix	56.8561 10.8863	80.3557 13.3984	112.8846 14.7869	153.4868 18.8882
Maximum difference in values from original matrix	5.977×10^{-13}	1.157×10^{-12}	5.506×10^{-12}	2.375×10^{-11}

The 'buf' field matrix set, Table 36, behaved similar to the Baseline and Network 2 and Network 3. There were a very low number of values per matrix, Table 35. This low value count yielded lower eigenvalues per matrix with most of their values being greater than one. The Maximum compressions for all time frames allowed for a reduction in the number of values by 48.6%, 22.7%, 20%, and 39.8% respectively.

Table 35. Uncompressed matrix using just 'buf' field - Network 4

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	523.6413	1047.283	2094.565	4189.13
Mean # values per matrix	31.4222 7.1574	47.8186 9.0189	72.7890 9.9570	102.6308 12.3360
Mean # of ports used per matrix	46.1353 8.0303	60.1134 8.9307	80.0249 9.4363	104.2897 11.6447
Mean # of Eigenvalues per matrix	6.0345 3.0863	10.9662 3.9263	17.5457 6.6753	29.6788 11.2040
Mean # of Eigenvalues < 1 per matrix	4.9411 2.3941	8.1877 2.2762	10.5856 1.8484	11.8841 1.8111
Maximum % compression	93	90	88	86

Table 36. Compressed matrix using just 'buf' field - Network 4

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	2824	6052.6	1019.3	1371.2
Mean # values < 1 at 50% compression per matrix	30.4051 7.0335	46.8495 8.9392	71.8563 9.9344	101.7533 12.3362
Maximum difference in values from original matrix	0.0025	0.0029	0.0058	0.0099
Mean # of values from SVD at Maximum	269.1668	809.4395	1676.3	2522.5
Mean # values < 1 at Maximum SVD per matrix	26.0928 5.5041	45.2327 8.2209	71.5066 9.7233	101.1623 12.1644
Maximum difference in values from original matrix	0.0648	0.0521	0.0075	0.041

Due to small amount of UDP packets, as seen in Table 37, for this dataset there is no need for further analysis of only UDP packets.

Table 37. Uncompressed matrix using just 'ulen' field - Network 4

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values per matrix	0.0736 0.2707	0.1401 0.3659	0.2542 0.4704	0.3957 0.5590

The combined matrix set for Network 4 also behaved similarly to the singular 'win' matrix set shown in Table 38 Table 39. The Maximum compressions failed to

yield fewer values than the original data. This continues the trend seen in the other network datasets.

Table 38. Uncompressed matrix using combined 'win' and 'ulen' fields - Network 4

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # of values from original data	654.5516	1309.103	2618.207	5236.413
Mean # values per matrix	56.9297 10.8928	80.4958 13.4118	113.1387 14.8042	153.8825 18.8931
Mean # of ports used per matrix	46.1353 8.0303	60.1134 8.9307	80.0249 9.4363	104.2897 11.6447
Mean # of Eigenvalues per matrix	22.1719 8.4630	36.3369 11.3659	50.2757 15.3484	70.1457 19.9229
Mean # of Eigenvalues < 1 per matrix	14.6317 3.5093	18.2982 3.8868	21.6080 4.1203	24.9553 4.4569
Maximum % compression	82	78	74	71

Table 39. Compressed matrix using combined 'win' and 'ulen' fields - Network 4

	0.25 sec	**0.5 sec**	**1.0 sec**	**2.0 sec**
Mean # values < 1 at 50% SVD per matrix	6657.6	10007	14093	18117
Mean # values < 1 at 50% compression per matrix	56.9297 10.8928	80.4958 13.4118	113.1387 14.8042	153.8825 18.8931
Maximum difference in values from original matrix	5.977×10^{-13}	1.157×10^{-12}	5.506×10^{-12}	2.384×10^{-11}
Mean # of values from SVD at Maximum	1549	2981.7	5474.5	8962.8
Mean # values < 1 at Maximum SVD per matrix	56.9297 10.8928	80.4958 13.4118	113.1387 14.8042	153.8825 18.8931
Maximum difference in values from original matrix	5.977×10^{-13}	1.157×10^{-12}	5.506×10^{-12}	2.384×10^{-11}

Overall Summary:

Network 4 dataset test produced similar results to the Baseline and Network 1 and Network 3. While the 'buf' field was able to decrease the overall number of values, the 'win' and combination 'win' and 'ulen' matrix sets did not. The other data fields did not yield few values than the original dataset.

4.2. Investigative Questions Answered

Going back to the original investigative question, "using SVD compression is the size of the reduced decomposed data less than the size of the original?" The answer to this question is the algorithm is inconsistent. For most of the datasets the 'win' field did not produce a lower amount of values than the amount of values needed to create the matrix. The 'buf' field did produce a lower amount of values, with the exception of Network 1. Analyzing the difference between the two fields in Network 1 shows the minimum values in the 'buf' field are much lower than the minimum values of the 'win' values by factor of 1000 between these matrices. When compared to 'buf' fields from other Networks, we see that quantity of values per matrix also increased during this dataset. This led to an increase in eigenvalues, and a decrease in the Maximum compression.

The second question, "How much can network traffic data be reduced by using SVD before the data values change?" is partially answered. For this specific size matrix the average amount of compression varied based on time frame and number of values per matrix. The amount of Maximum compression decreased as time frame, and as a result matrix data, increased. The value for Maximum compression with SVD compression is greater than 50% for every evaluation where the individual values do not change by more than a value of one. This is due to the sparse nature of the matrices created. Since more than 50% of the matrix is empty, the method of measuring compression is always greater than 50%.

However this leads into the final question: "Is the Maximum compression enough to reduce the overall amount of data before a change in the individual values occurs?"

The answer varies in the experiment. For the entire 'win' field compressions the number of values failed to reduced the number of values in the decompositions to less than amount of the original data. However, most of the 'buf' field compressions decreased the number of values. Network 1 was the only network that did not yield a reduced number values for the 'buf' field.

Examining this dataset relative to the others, Figure 5 shows the total number of cells containing a value per matrix for the 'buf' element across all the networks. The graph clearly shows the number of values per matrix in the Network 1 dataset is higher than all the others. When comparing the average value in each cell, Figure 6, we see that the value of any element did not determine if the number of values in the decomposition, but the overall total number of values in the individual matrix.

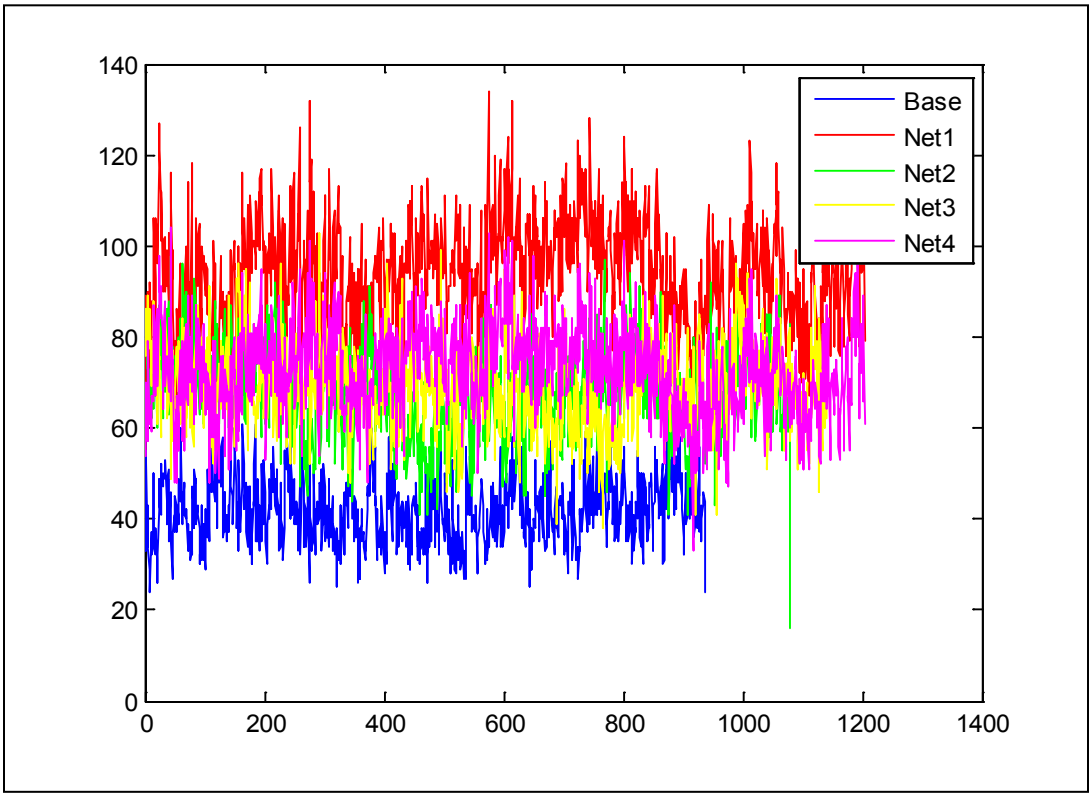

Figure 5: Average # of Values per Matrix of 'buf' field, 1.0 sec time frame

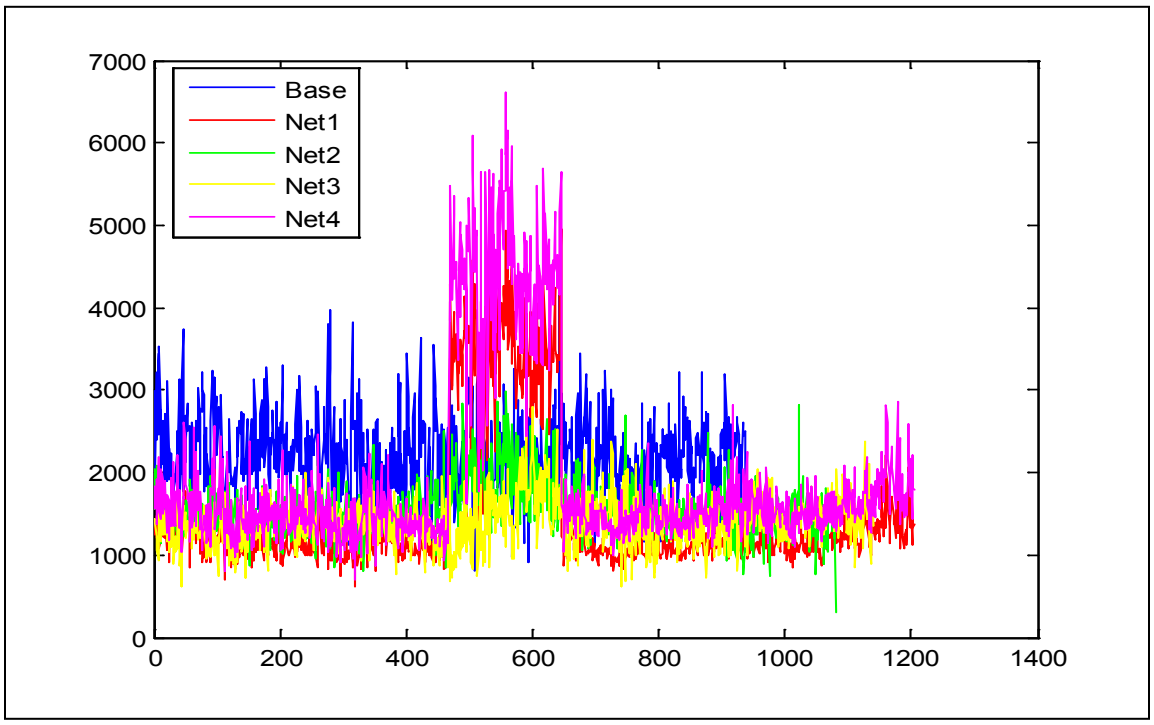

Figure 6. Average Cell Value per Matrix of 'buf' field, 1.0 sec time frame

When looking at Table 6, the reason the SVD compression succeeded in the 2.0 second time frames is most of the new packets added to the matrix were combined in cells that already contained a non-zero value. The number of original values would increase at a steady rate based on the count. Based on this, if time frame were increased the compression algorithm would eventually hold. However, the combining of too many field values in the individual cells could distort activity that happens during that time frame.

The data gathered from this experiment shows the following trends. Compression percentage is dependent on the number of significant eigenvalues in each matrix. However, the amount of compression was not equal to the number of significant eigenvalues. The number of significant values kept from compression was approximately twice the value of the mean number of eigenvalues plus one standard deviation. This

shows that even though there was a division in values of eigenvalues for each matrix, the highest small valued eigenvalues were not as insignificant as their values for maintaining individual value integrity.

Another trend observed throughout this experiment is that the low number of data values in the original matrix, without being compressed, allows for a lower amount of data values than the original data. Comparing the first two rows of each table representing the uncompressed matrix verified this. The difference in values actually decreases with the increase in time frame. This decrease can be explained by trend of data occupying the same space inside the matrix, e.g. the matrix cell that contains multiple packets, combines the values of all those packets into the same cell in the matrix. This experiment allowed the combining of data to take place because the values represented bytes transferred over the measured time frame. This would not work for another type of network traffic element, such as the sequence number in the 'seq1' or 'seq2' fields. These fields cannot be combined in this simple manner. However, based on the algorithm of placement the number of values per matrix would not change, as the same number of packets would place values into the matrix, only the value itself.

V. Conclusions and Recommendations

5.1. Conclusions of Research

This research concludes that using SVD compression for the reduction of data in network monitoring traffic does not sufficiently decrease the volume of data for any specific data type. While certain data fields can reduce the amount of data, the algorithm described in this thesis does not hold for all fields of network monitoring traffic. Appendix B shows the results of at two sample t-test performed between the number of values of original data per matrix and the number of values needed at Maximum compression at a 95% confidence interval. The results show that almost all datasets do not have equal means confirming the results in Chapter 4. The sole outlier was the 'win' field at 2.0 sec interval in the Baseline dataset. The t-test for this outlier yielded a p-value of 0.6503 meaning the difference between the two samples cannot be rejected given the mean and variance of the two samples. All other p-values were much less than 0.001.

The data suggests that the network monitoring traffic SVD compression algorithm only works with a small number of values per matrix relative to the number of original inputs. The experiment illustrates that the number of matrix cells containing values grows at a slower rate than the number of packets contributing data to any individual matrix. This confirms that most of the data inserted into the matrix was done by combining the data from multiple packets. Dependent upon the needs of the network monitoring tool or administrator, the amount of data combination may be irrelevant. To allow for the option of less data combining, the 0.25 and 0.5 second time frames indicate that SVD compression does not work for all fields.

5.2. Significance of Research

According to George Gilder's Law the total bandwidth of communication systems triples every twelve months. This law of technology has held true, and the amount of network traffic globally has steadily increased. Gordon Moore's Law states that the processing power of a microchip doubles every 18 months. These two laws are related as one cannot continue without the other. From this increasing supply of data comes the need to monitor this data traffic. Technology is advancing at rate where individual data values are simple to store and transfer. This trend does not imply that measures to reduce this data should not be explored.

For example, the increasing use and adaptation of mobile technology has created a drop from these two laws traditional means of measurement. While these two laws are still relevant to the new mobile technology, they are dealing with limited resources for each devise. Chief amount these limitations is power, as the amount of power a mobile devise has limits the amount of processor, amount of memory in a devise, or strength of wireless signal. Without changing the amount of power a devise has or uses, a method used to reduce the amount data stored or sent to a mobile devise would conserve the resources.

This research showed that data reduction using SVD compression does not allow for an overall reduction in data. The individual success of the 'buf' field for most networks demonstrates the potential for SVD compression. However, SVD compression in most of the experiments produced more values than the number used from the original data. Unless the ratio of original data to the number of values in a matrix can be

controlled, this method of data reduction is not beneficial for reducing total number of data values.

5.3. Recommendations for Future Research

Future research along the path of network monitoring data reduction could include multiple different aspects. A potential aspect to maintain from this experiment is the matrix model for parsing network traffic data. The use of data in matrix format is a versatile capability. Once the data is in matrix form, any variety of manipulations and calculations can be performed upon it. Though the method of reduction did not meet the parameters set in this experiment. Further experiments could find this method useful for individual tools, if the need for change in individual values were not as strict.

Future research in this area should incorporate some of the following suggestions. The use of a more modern dataset, matrix population by IP address instead of port, and the use of a difference matrix compression/reduction method are three of the most significant contributions to this field of research.

As mentioned before, the values and data captured in these datasets were recorded in 1996. Modern datasets could contain more specific data either per packet or per connection. The use of a connection or flow based data measurements instead of packets, as mentioned in Chapter 2, could result in more favorable outcomes for the SVD compression.

Changing the matrix population scheme from port based to IP based is another modification for future research. This change would be dependent upon how the recipient of the data interprets the data. Both IP and port could be incorporated to add a

further diversification, conversely as shown in this experiment, the SVD compression method performs only when the matrix is very sparsely filled relative to the amount of original data. Incorporating both address and port would increase unique values but also increase the amount of data generated from SVD compression.

Finally this experiment could be conducted examining another matrix reduction or compression methods. This experiment examined only one image compression scheme. There are other compression methods available which would require conversion to and from a specified format, such as GIF, JPEG, or PNG. This does not include proprietary image compressions of the mentioned formats.

Network monitoring traffic compression is a relevant subject for exploration. Networks will only grow larger and the amount of data transmitted will continue to increase exponentially. In order to monitor network traffic in an efficient and resourceful manner, one must find a way to compress or represent the larger amounts of data.

Appendix A. Raw dataset sample

time	src_addr	src_port	dest_addr	dest_port	flag	seq1	seq2	ack	win	buf	ulen	op
38141.5	1	7000	2	7001	U						148	
38141.51	3	20	2	3421	.	1811081902	1811082414	366784001	9216	512		
38141.52	3	20	2	3421	.	512	1024	1	9216	512		
38141.52	4	80	2	2609	.			438528422	9112			(DF)
38141.52	5	25	2	1362	F	266688477	266688477	580609140	4096	0		
38141.52	6	119	2	4305	P	1536324798	1536324811	1647611259	61440	13		
38141.52	2	1362	5	25	.			1	4096			
38141.53	7	27383	2	1826	.	53897419	53897931	318976001	4096	512		
38141.54	2	2611	8	80	.			786087235	16384			
38141.54	2	2593	9	80	.			1163562564	16384			
38141.54	2	1826	7	27383	.			0	16384			
38141.54	7	27383	2	1826	.	512	1024	1	4096	512		
38141.54	4	80	2	2608	.			438464421	9112			(DF)
38141.54	2	4305	6	119	P	1	38	13	4096	37		
38141.54	2	1826	7	27383	.			1024	16384			
38141.55	3	20	2	3421	.	1024	1536	1	9216	512		
38141.55	3	20	2	3421	.	1536	2048	1	9216	512		
38141.55	2	53	10	53	X							: 33259 [b2&3=0x30] (32)
38141.56	3	20	2	3421	.	2048	2560	1	9216	512		
38141.56	2	2621	4	80	S	440192000	440192000		16384	0		
38141.56	7	27383	2	1826	.	1024	1536	1	4096	512		
38141.57	7	27383	2	1826	.	1536	2048	1	4096	512		
38141.57	2	1826	7	27383	.			2048	16384			
38141.57	7	27383	2	1826	.	2048	2560	1	4096	512		
38141.58	11	80	2	2613	.	517245043	517245579	439232285	8292	536		(DF)
38141.58	2	119	12	4803	.			806839108	23049			
38141.58	11	80	2	2613	.	536	1072	1	8292	536		(DF)
38141.58	13	2845	2	25	P	1407111916	1407111962	581376195	9112	46		(DF)
38141.58	2	25	13	2845	.			46	4096			
38141.58	2	2613	11	80	.			1072	16384			
38141.59	11	80	2	2613	.	1072	1608	1	8292	536		(DF)
38141.59	1	7000	2	7001	U						148	
38141.59	4	80	2	2620	P	953393175	953393278	439936419	9112	103		(DF)
38141.59	4	80	2	2620	F	103	103	1	9112	0		(DF)
38141.59	2	2620	4	80	.			104	16281			
38141.59	2	2620	4	80	F	1	1	104	16384	0		
38141.6	11	80	2	2324	.	517348953	517349489	395584281	8296	536		(DF)
38141.6	8	80	2	2611	.	1	537	0	8576	536		(DF)
38141.61	8	80	2	2611	.	537	1073	0	8576	536		(DF)
38141.61	2	2611	8	80	.			1073	16384			
38141.61	8	80	2	2611	.	1073	1609	0	8576	536		(DF)
38141.61	6	119	2	4305	.			38	61440			
38141.61	6	119	2	4305	P	13	26	38	61440	13		
38141.62	3	20	2	3421	.	2560	3072	1	9216	512		
38141.62	3	20	2	3421	.	3072	3584	1	9216	512		

Appendix B. Two sample T-test Results

Table 40. H-values for Two sample T-test

	0.25sec	0.5sec	1.0sec	2.0sec
Baseline				
Win	1	1	1	1
Buf	1	1	1	1
Win Ulen	1	1	1	0
Network 1				
Win	1	1	1	1
Buf	1	1	1	1
Win Ulen	1	1	1	1
Network 2				
Win	1	1	1	1
Buf	1	1	1	1
Win Ulen	1	1	1	1
Network 3				
Win	1	1	1	1
Buf	1	1	1	1
Win Ulen	1	1	1	1
Network 4				
Win	1	1	1	1
Buf	1	1	1	1
Win Ulen	1	1	1	1

Two-sided t-test was conducted with a confidence interval of 95%

Values of 1 are interpreted as the means of the two data sets are not statistically similar given mean and variance.
Values of 0 are interpreted as the means cannot be seen as statistically different given mean and variance.

Bibliography

[1] The 1998 intrusion detection off-line evaluation plan. MIT Lincoln Lab., Information Systems Technology Group. http://www.11.mit.edu/IST/ideval/docs/1998/id98-eval-11.txt, 1998.

[2] Abdullah, Kulsoom, et al. "Ids rainstorm: Visualizing ids alarms." IEEE Workshop on Visualization for Computer Security. Vol. 2005. 2005.

[3] Andrews, H.C., C.L. Patterson. Singular value decomposition (SVD) image coding. IEEE Transactions on Communications 24 (1976), pp. 425–432.

[4] Best, Daniel M., et al. "Real-time visualization of network behaviors for situational awareness." Proceedings of the Seventh International Symposium on Visualization for Cyber Security. ACM, 2010.

[5] Casas, Pedro, Johan Mazel, and Philippe Owezarski. "Knowledge-independent traffic monitoring: Unsupervised detection of network attacks." Network, IEEE 26.1 (2012): pp. 13-21. 2012.

[6] Choi, Hyunsang, Heejo Lee, and Hyogon Kim. "Fast detection and visualization of network attacks on parallel coordinates." computers & security 28.5 (2009): 276-288.

[7] D'Amico, Anita D., et al. "Visual discovery in computer network defense." Computer Graphics and Applications, IEEE 27.5 (2007): 20-27.

[8] Eckart, C. and G. Young. The approximation of one matrix by another of lower rank, Psychometrika,1 (1936), pp. 211–218.

[9] Estrin, Deborah, et al. "Network visualization with nam, the vint network animator." Computer 33.11 (2000): pp. 63-68.

[10] Fischer, Fabian, et al. "Large-scale network monitoring for visual analysis of attacks." Visualization for Computer Security (2008): pp. 111-118.

[11] Girardin, Luc. "An Eye on Network Intruder-Administrator Shootouts." Workshop on Intrusion Detection and Network Monitoring. 1999.

[12] Goodall, John R. "Visualization is better! a comparative evaluation." Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on IEEE, 2009.

[13] Goodall, John R., and Mark Sowul. "VIAssist: Visual analytics for cyber defense." Technologies for Homeland Security, 2009. HST'09. IEEE Conference on IEEE, 2009.

[14] Grinstein, Georges G., et al. "Benchmark development for the evaluation of visualization for data mining." Information visualization in data mining and knowledge discovery, 129-176, 2002.

[15] Harrison, Lane, and Aidong Lu. "The future of security visualization: Lessons from network visualization." Network, IEEE 26.6, pp. 6-11, 2012.

[16] Index, Cisco Visual Networking. "Global mobile data traffic forecast update, 2012–2017, http://www.cisco.com/enUS/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html, 2013.

[17] Jackson, David Jeff and Sidney Joel Hannah. "Comparative analysis of image compression techniques." System Theory, 1993. Proceedings SSST'93, Twenty-Fifth Southeastern Symposium on IEEE, 1993.

[18] Keim, Daniel A. "Information visualization and visual data mining." IEEE transactions on Visualization and Computer Graphics 8.1, pp. 1-8, 2002.

[19] Lee, Wenke, Salvatore J. Stolfo, and Kui W. Mok. "A data mining framework for building intrusion detection models." Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on IEEE, 1999.

[20] Mansmann, F., F. Fischer, D.A. Keim, and S.C. North. “Visual support for analyzing network traffic and intrusion detection events using TreeMap and graph representations". Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology, pp. 3, 2009.

[21] Marty, R. “Applied security visualization.” Addison-Wesley, 2009.

[22] North, C., Kerren, A., Stasko, J., and Fekete J. “Information Visualization: Human-Centered Issues in Visual Representation, Interaction, and Evaluation.” Springer-Verlag. 2008.

[23] Orebaugh, Angela, Gilbert Ramirez, and Josh Burke. “Wireshark and Ethereal network protocol analyzer toolkit.” Syngress Media Incorporated, 2007.

[24] Ramadas, Manikantan, Shawn Ostermann, and Brett Tjaden. "Detecting anomalous network traffic with self-organizing maps." Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2003.

[25] Rangarajan, Anand. "Learning matrix space image representations." Energy Minimization Methods in Computer Vision and Pattern Recognition. Springer Berlin Heidelberg, 2001.

[26] Saydjari, O. Sami. "Cyber defense: art to science." Communications of the ACM 47.3, pp. 52-57, 2004.

[27] Shalf, John, and E. Wes Bethel. "The grid and future visualization system architectures." Computer Graphics and Applications, IEEE 23.2 (2003): pp. 6-9, 2003.

[28] Shiravi, H., A. Shiravi, and A. Ghorbani. “A survey of visualization systems for network security". Visualization and Computer Graphics, Transactions on IEEE, 11, 2011.

[29] Swanson, Iain. “Malware, Viruses and Log Visualization.” In Australian Digital Forensics Conference. Paper 54, 2008.

[30] Tesone, Daniel R., and John R. Goodall. "Balancing interactive data management of massive data with situational awareness through smart aggregation." Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium, 2007.

[31] Waldemar, P., T.A. Ramstad, Image compression using singular value decomposition with bit allocation and scalar quantization, in: Proceedings of NORSIG Conference, 1996, pp. 83–86. 1996.

[32] Ware, C. “Information Visualization: Perception for Design.” Morgan Kaufmann Publishers Inc., 2000.

[33] Whitaker, Robert Bruce. "Applying Information Visualization to Computer Security Applications." 2010.

[34] Yost, Beth, and Chris North. "The perceptual scalability of visualization." Visualization and Computer Graphics, IEEE Transactions on 12.5 (2006): pp. 837-844. 2006.

[35] Zage, Dolores M., and Wayne M. Zage. “Intrusion Detection System Visualization of Network Alerts.” 2010.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
23-08-2012	Master's Thesis	August 2012 – March 2014

TITLE AND SUBTITLE

Network Monitoring Traffic Compression Using Singular Value Decomposition

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Steven N. Feigh, MAJ, USA

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENG)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-14-M-27

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Intentionally left blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRUBTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
With increasing magnitude of computer network activity, the ability to monitor all network traffic is becoming strained. The need to represent large amounts of data in smaller forms is essential to continued growth of network monitoring tools and network administrators' capabilities. Network monitoring captures many different measurements of the data flowing through the network. This thesis introduces a new method of sending network traffic monitoring data that reduces the overall volume of data from the traditional method of packet capture. By populating a matrix with specific data values in a sparse format, this experiment reduces the data using singular value decomposition (SVD) compression. Matrices were populated using network monitoring datasets from 1996 Information Exploration Shootout (IES). The data populated into the matrices was varied along time frame and data field to determine if the SVD compression algorithm reduced the quantity of original data values. Results indicated that the quantity of data varies dependent on the volume of the data field chosen. The matrix population method was based on port values to allow combining values within the matrix cells. The results trended to a successful reduction of data if the time frame is increased significantly.

15. SUBJECT TERMS
Network Monitoring, Network Traffic

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kennard R. Laviers, AFIT/ENG
U	U	U	UU	75	19b. TELEPHONE NUMBER *(Include area code)* (937) 785-3636, (Kennard.Laviers@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18